PyTorch Project to Build a LSTM Text Classification Model

Overview

Long short-term memory(LSTM) is a recurrent neural network architecture(RNN) used in the field of deep learning. These architectures are capable of learning long-term dependencies faced by recurrent neural networks. LSTMs have feedback connections which makes them different from the traditional feed-forward neural networks. LSTMs are particularly good at text data, speech, and time series.

In this project, an LSTM model for classifying the review of an app on a scale of 1 to 5 based on the feedback has been built in PyTorch. If you haven't visited already, here is the previous project of the series Build a CNN Model with PyTorch for Image Classification

Aim

- To understand the working of LSTM
- To classify the review of an app on a scale of 1 to 5 using LSTM

Data Description

The dataset contains the reviews and the ratings of the app. The dataset has a score column and content column. The score columns have a number range between 1 to 5 based on the content column.

Tech Stack

- ➔ Language: Python
- ➔ Libraries: pandas, tensorflow, matplotlib, sci-kit learn, nltk, numpy, pytorch

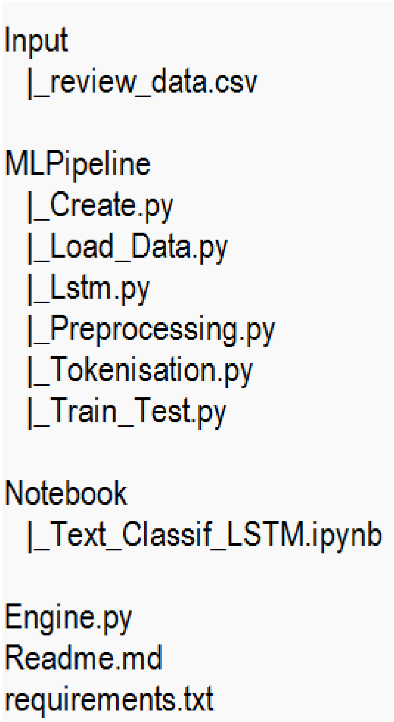
Approach

- Data Preprocessing
  - Lowering Text, removing punctuation, removing links
  - Balancing classes
  - Tokenizing the text
  - Scaling

- Model Training
  - Training LSTM model in PyTorch

- Model Evaluation
    - Evaluation of model on test data

Modular Code Overview



```
Input
  |_review_data.csv

MLPipeline
  |_Create.py
  |_Load_Data.py
  |_Lstm.py
  |_Preprocessing.py
  |_Tokenisation.py
  |_Train_Test.py

Notebook
  |_Text_Classif_LSTM.ipynb

Engine.py
Readme.md
requirements.txt
```

Once you unzip the pytorch_rnn.zip file, you can find the following folders within it.

1. Input
2. ML_Pipeline
3. Notebook
4. Engine.py
5. Readme.md
6. requirements.txt

1. The Input folder contains the data that we have for analysis. In our case, it contains review data

2. The Notebook folder contains the jupyter notebook file of the project

3. The ML_pipeline is a folder that contains all the functions put into different python files, which are appropriately named. These python functions are then called

inside the Engine.py file

4. The requirements.txt file has all the required libraries with respective versions. Kindly install the file by using the command **pip install -r requirements.txt**

5. **All the instructions for running the code are present in Readme.md file**

Takeaways

1. What is PyTorch?
2. PyTorch vs Tensorflow
3. Limitations of current algorithms
4. What is a text analytics and the need for text analytics?
5. What is RNN and the need for RNN?
6. Architecture of RNN
7. What is LSTM?
8. Architecture of LSTM
9. Working of LSTM
10. Application of LSTM
11. Tokenization of text
12. Lemmatization of text
13. Converting data to tensors
14. Handling class imbalance
15. Building LSTM model in PyTorch